Neural Network Analysis of Dimuon Data within CMS

Shannon Massey
University of Notre Dame

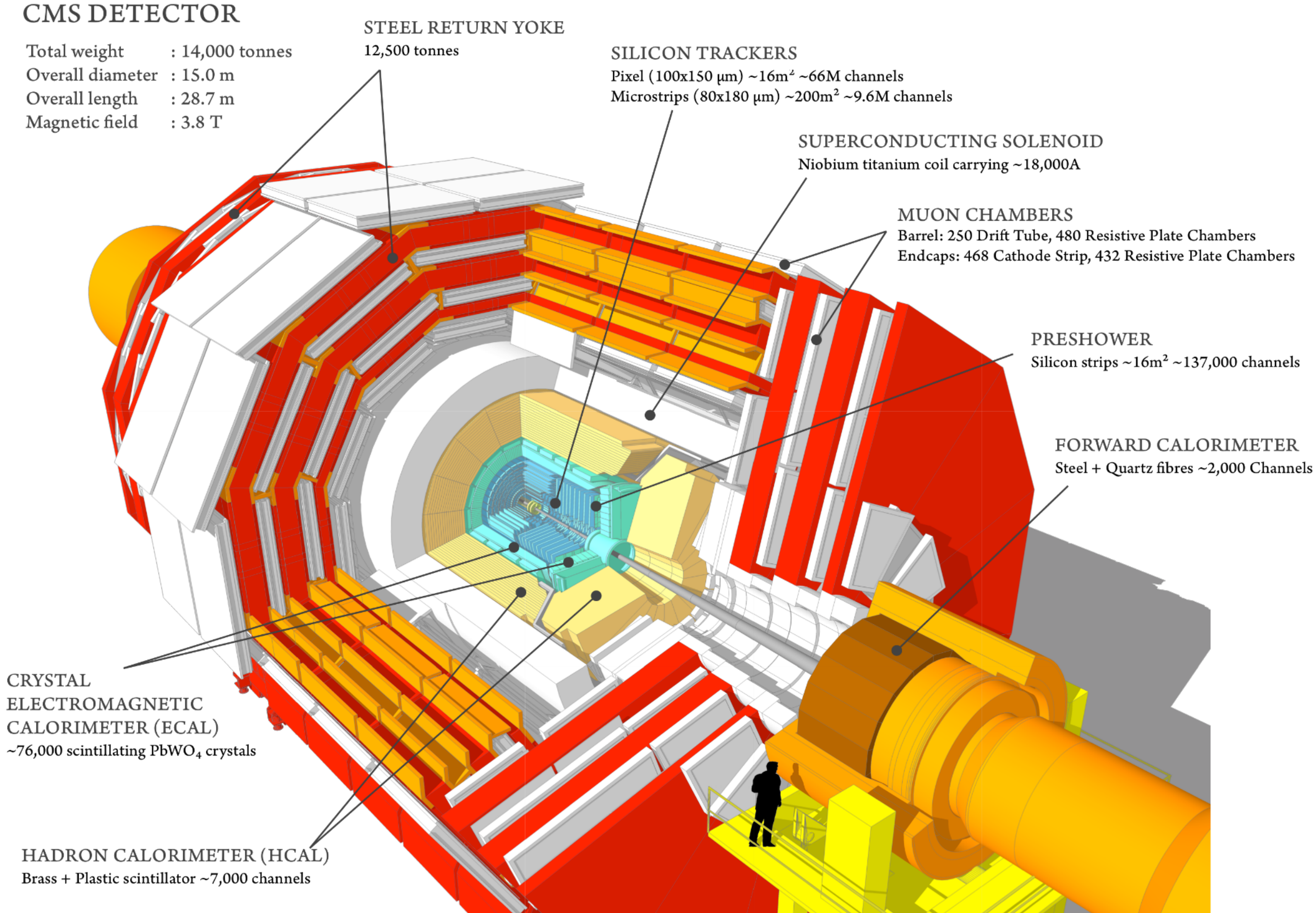
Overview

- The CMS Detector
- Muons and Event Display
- Classification and Machine Learning
- Signal and Background Discrimination in Dimuon Final State
- Summary
- Acknowledgements

The Compact Muon Solenoid (CMS) Detector

- General-purpose collider detector
- Modular design
- Four main parts
  - Tracker, Electromagnetic Calorimeter (ECAL), Hadron Calorimeter (HCAL), Muon System
- A large solenoid magnet
  - Operated a field of 3.8 Tesla
  - ~100,000 times the magnetic field of the Earth

Compact Muon Solenoid (CMS)

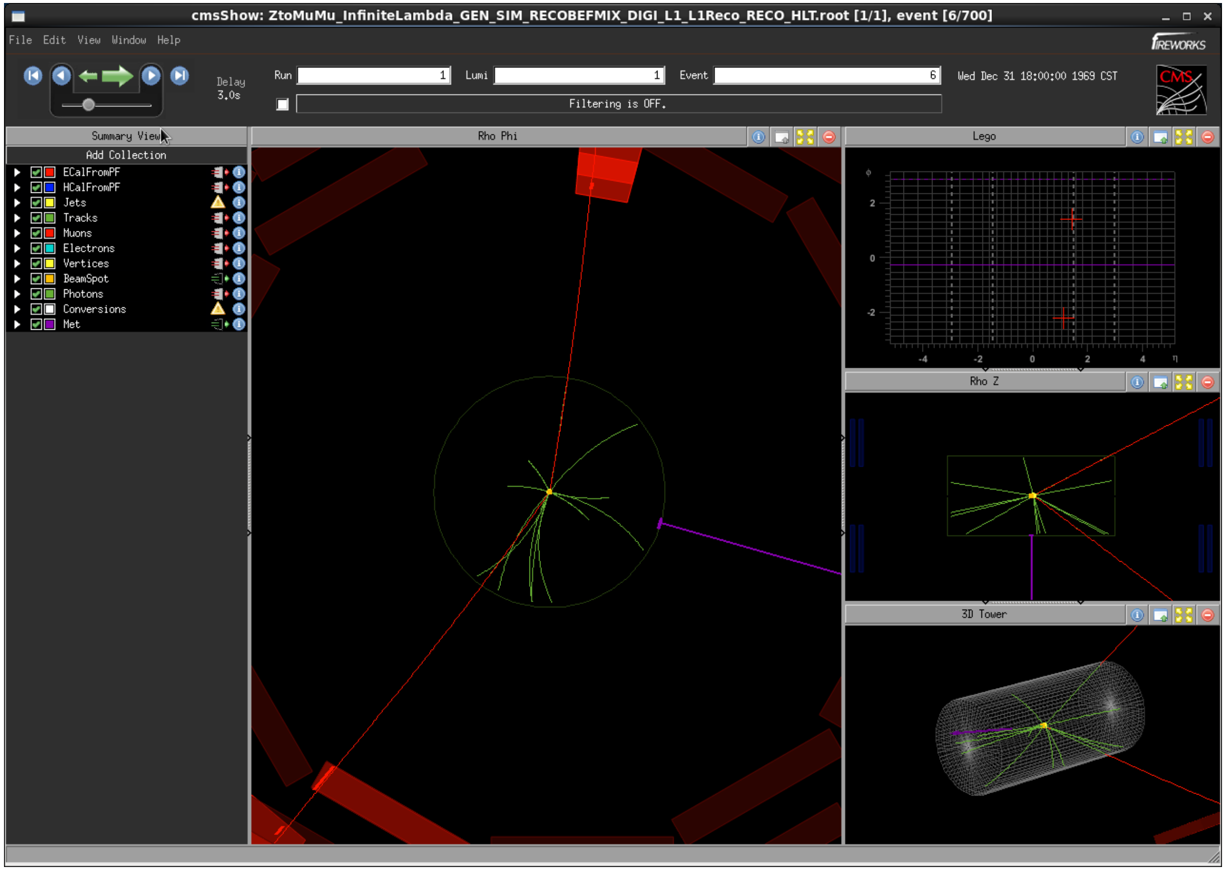


Muons

- CMS identifies and measures muons using silicon tracker and the outer muon system
- Produced in the decay of Standard Model and potential new particles
  - Example: $Z \rightarrow \mu\mu$, $H \rightarrow ZZ \rightarrow 4\mu$, $Z' \rightarrow \mu\mu$
- Charge: $\pm 1\ e$
- Mass of ~105.7 MeV/c^2
  - Mass of an electron = ~.51 MeV/c^2
- Half-Life of 2.2 μs
- Good to study since it can be identified easily and measure well
  - Also, smaller backgrounds in high p_T muon final states

Fireworks

- Event display through Fireworks (cmsShow)
- Ability to visualize reconstructed data



Binary Classification with Neural Networks

- Discrimination between signal and background events/objects
- Essential to find interesting and rare signal events within gigantic data sets
  - Analogous to finding a needle in a haystack
- Machine Learning algorithms can be used to classify data

Machine Learning

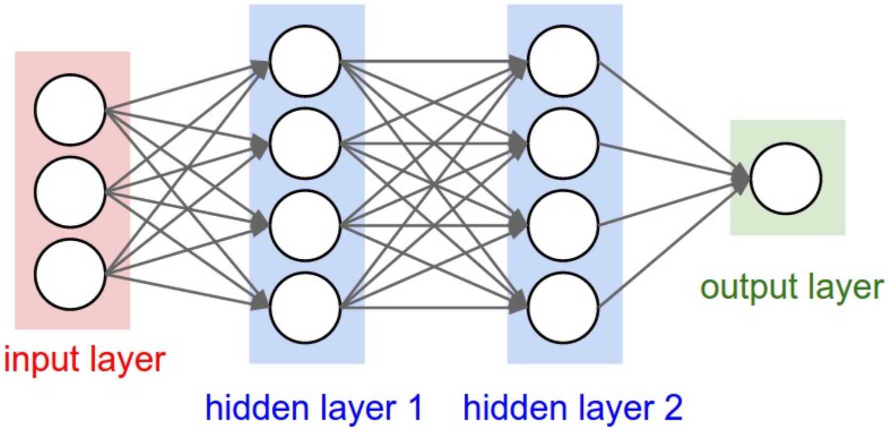
- Supervised Learning
  - Trained through numerous labelled examples
  - Examples: Siri, Image Recognition, Text Recognition (Spam), many others
- Unsupervised Learning
  - No labels are given to the learning algorithm in the examples
  - Must find structure in the inputs on its own
  - Used to: recognize patterns within data, categorize data

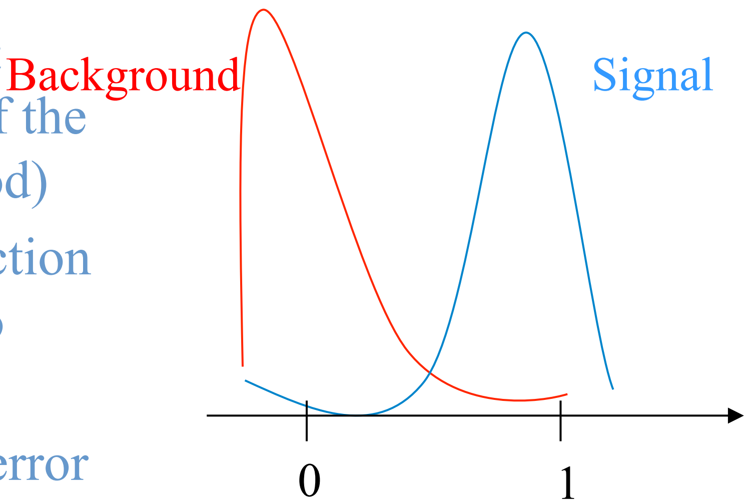
Toolkit for Multivariate Analysis (TMVA)

- Integrated into ROOT
- Includes many different multivariate classification algorithms
  - Fisher discriminants (linear discriminant analysis)
  - K-Nearest Neighbor
  - Boosted Decision and Regression Trees
  - Artificial Neural Networks (ANN)
- All algorithms are supervised learning
- My work focused on ANNs
  - More specifically Multilayer Perceptrons (MLPs)

Multilayer Perceptrons

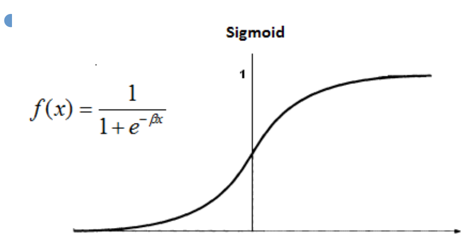
- Network of "hidden", simple neurons (perceptrons)
- Linked by feed-forward connections
- In order to learn from a set of inputs, TMVA MLPs use back-propagation (of errors)
  - To change the weights, we must compute the partial derivative of the weights (gradient descent method)
  - The derivatives give us the direction (+/-) the weights must change to reduce error
  - Goal: minimize miscalculation error

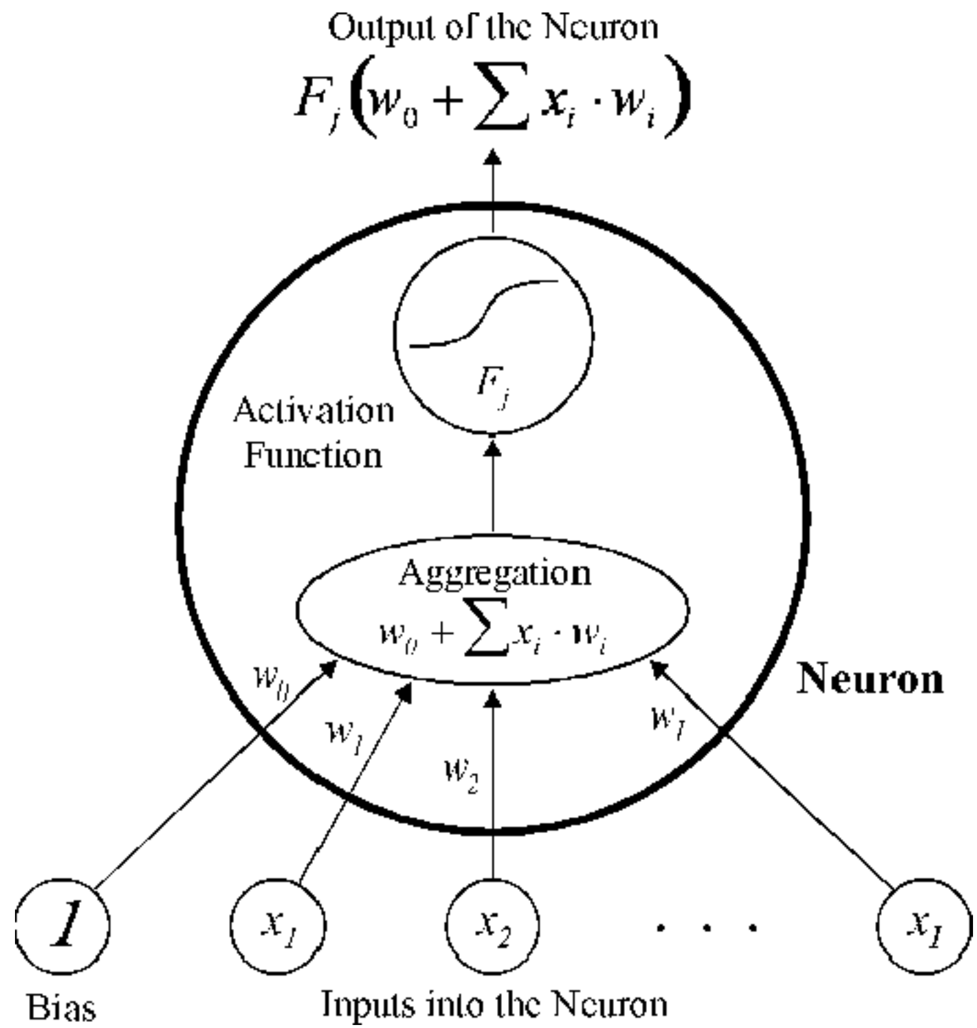




Multilayer Perceptrons

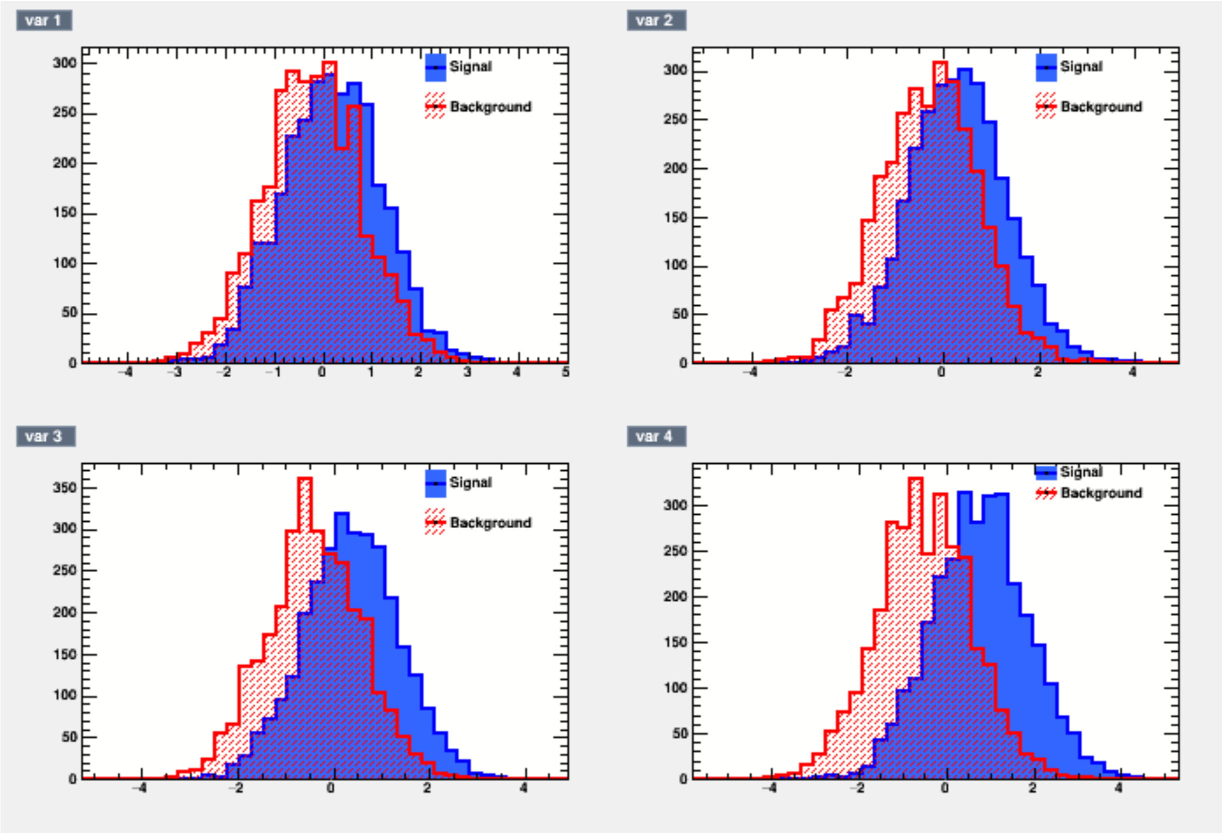
- In TMVA, activation functions for MLPs:
  - Linear: x
  - Sigmoid: $1/1+e^{-x}$
  - Tanh: $e^{2x}-1/e^{2x}+1$
- Sigmoid popular for binary classification
  - nd 1





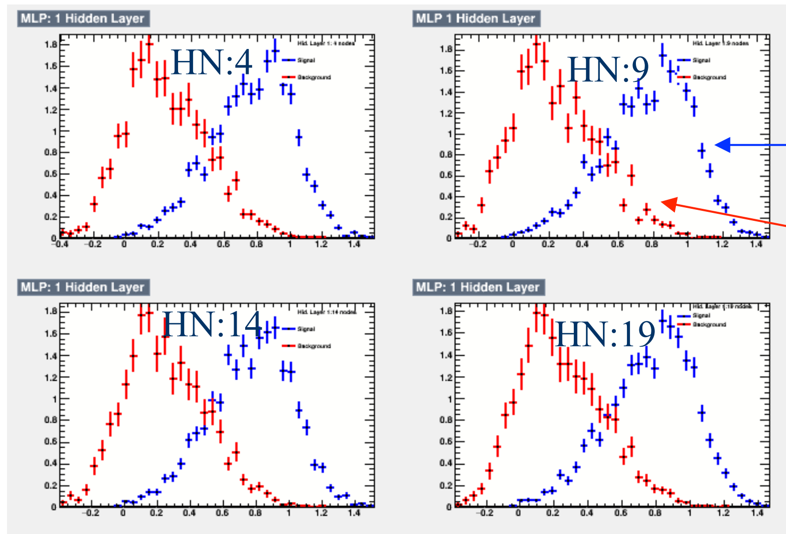
Toy Data within TMVA

- TMVA provides an example with a toy data set
  - Signal and Background for var 1, var 2, var 3, var 4



MLP Output from Toy Data

- MLP Attributes
  - Sigmoid activation function
  - 600 training cycles
  - 8 different MLPs
    - 1 and 2 Hidden Layers
    - 4, 9, 14, 19 Nodes
- Difficult to determine effectiveness of the algorithms by eye

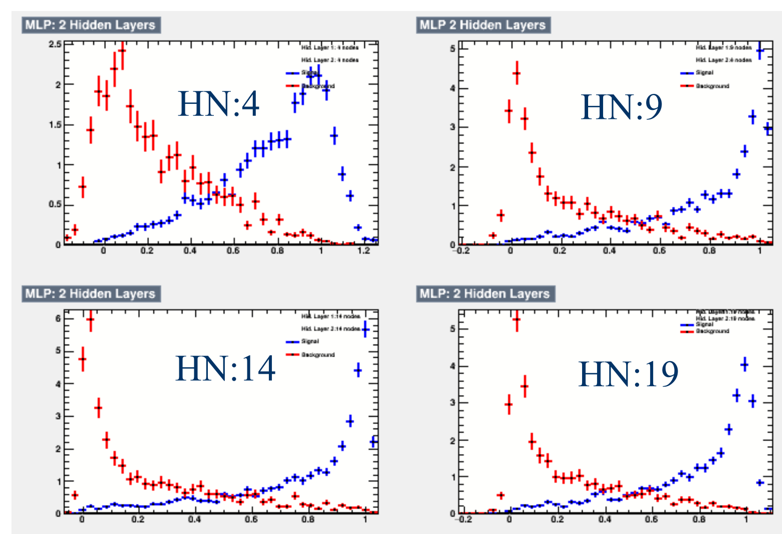


One Hidden Layer

Signal

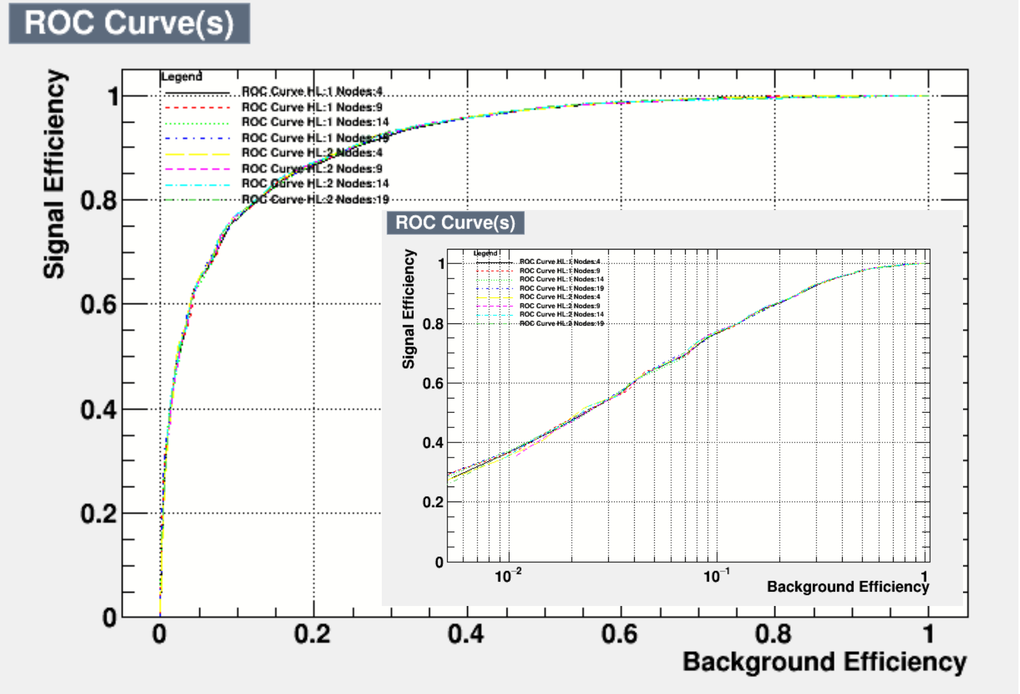
Background

Two Hidden Layers



MLP Analysis of Toy Data

- Receiver Operating Characteristic Curve (ROC Curve) useful in determining effectiveness of an algorithm
- Plots background at each x-value/total background vs. signal at each x-value/total signal
  - AUC of a perfect algorithm = 1
  - AUC of a completely random = .5



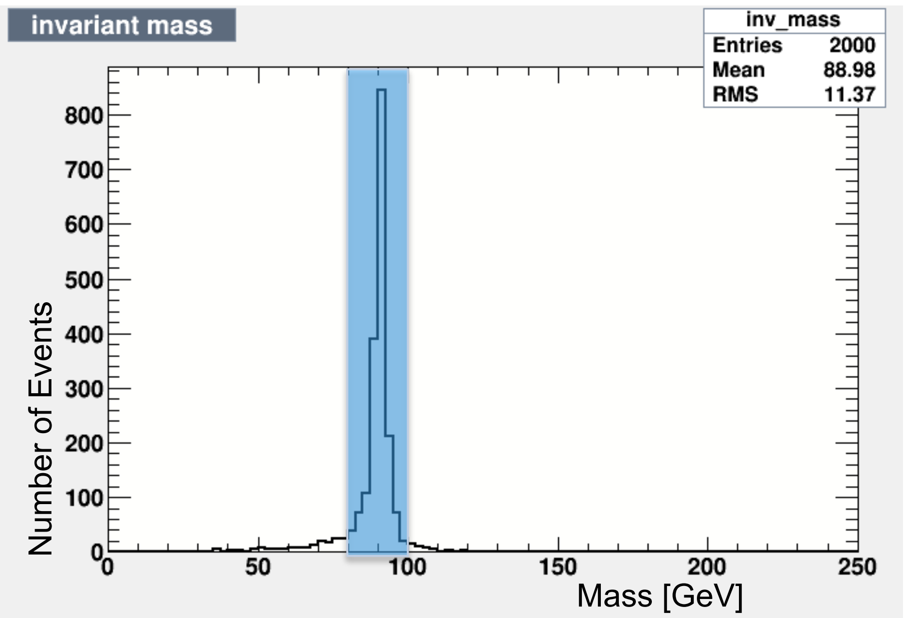
Toy Data: Area Under the Curve Table			
Total Hidden Layers:	Nodes in Layer One:	Nodes in Layer Two:	Area Under the ROC Curve (AUC)
1	4		0.9189
1	9		0.9194
1	14		0.9192
1	19		0.9195
2	4	4	0.9202
2	9	9	0.9190
2	14	14	0.9195
2	19	19	0.9200

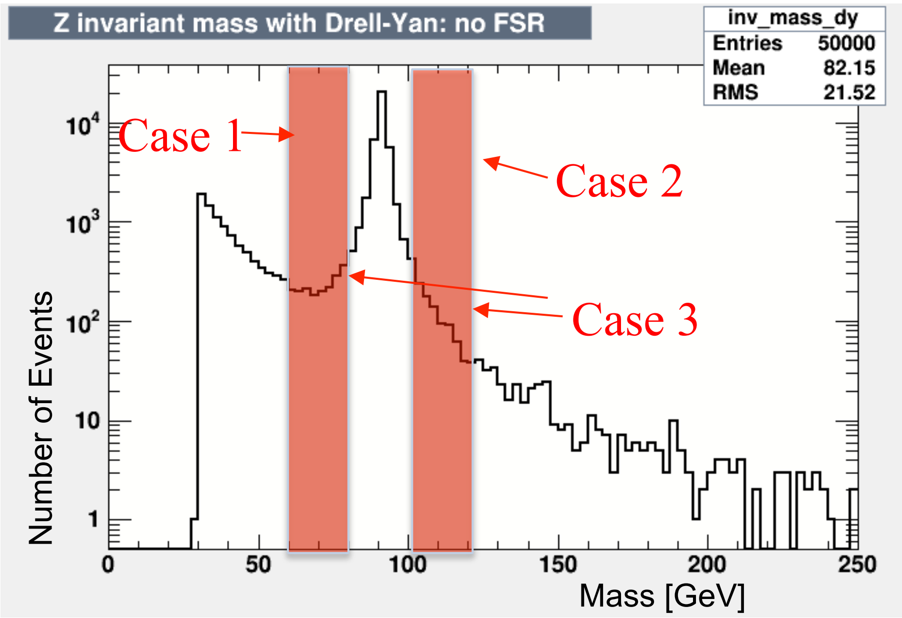
Event Reconstruction

- Many Thanks to Grace for dimuon samples!
- Applied a Python script with C++ analyzer to .root generation file
    - Select desired events (Muons)
    - Include necessary attributes (p_t , eta, phi)

Classification with Dimuons (Case 1)

- MLP implementation to discriminate Dimuons in Mass Peak and side-bands
  - Signal input: Kinematic Variables Dimuon Mass Peak
    - 80 GeV – 100 GeV (1,758 Events within the range)
  - Background input: surrounding Drell-Yan background
    - Mass: 60 GeV - 80GeV (1,867 Events within the range)





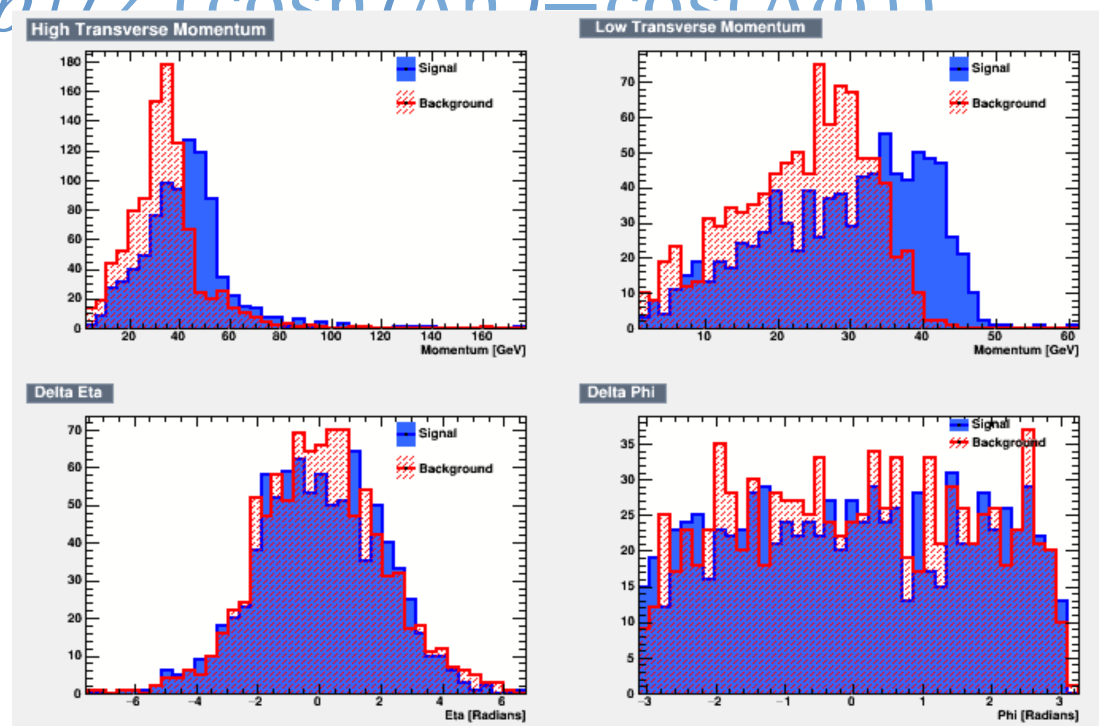
Classification of Dimuon Mass Peak

- Signal and Background inputs
- Four Variables included in the calculation of the dimuon mass
  - High Transverse Momentum (p_t), Low p_t, Delta Eta, Delta Phi

$$M_{\mu+\mu-}=\sqrt{2p_{t1}\,p_{t2}\,(\cosh(\Delta\eta)-\cos(\Delta\varphi))}$$

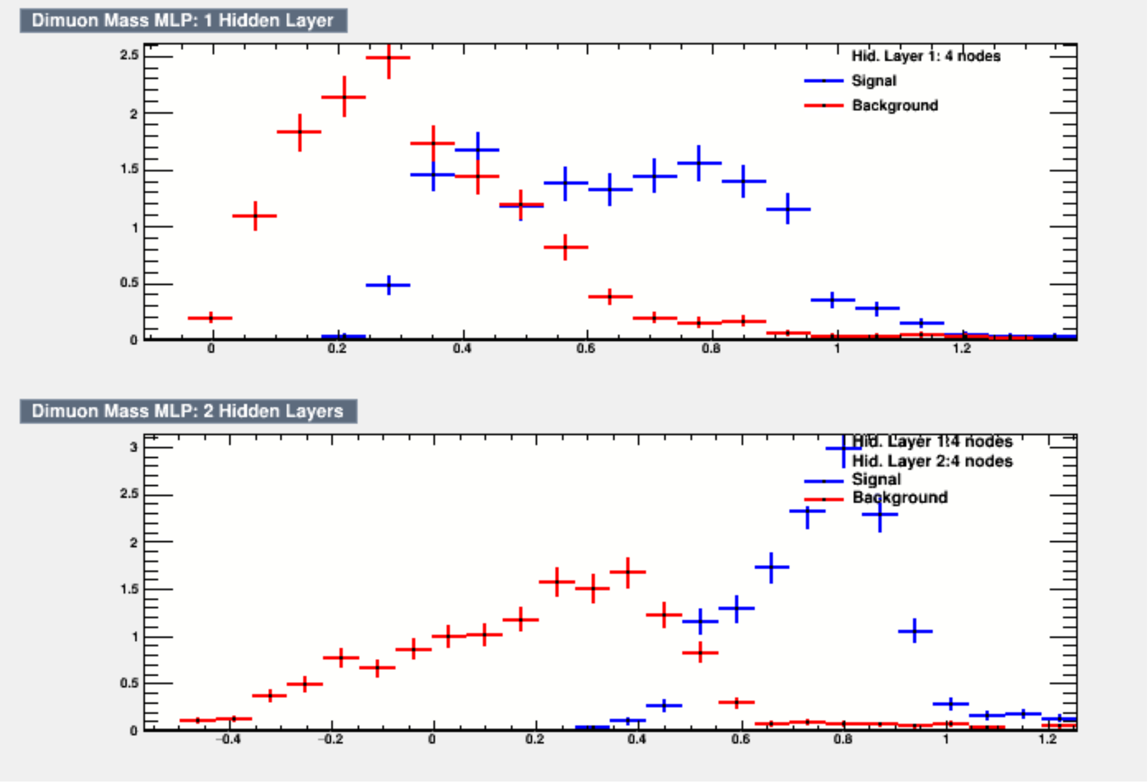
Case 1

  - Naturally not as separated as the toy data input variables



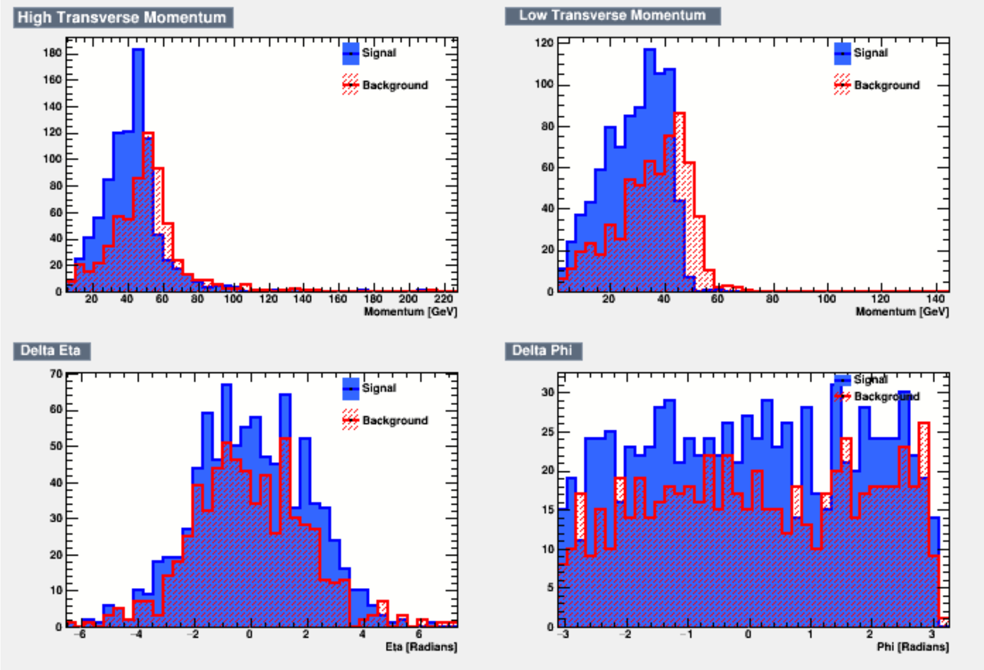
Classification (Case 1)

- MLP Attributes
  - Sigmoid activation function
  - 5,000 training cycles
  - 2 MLPs
    - 1 and 2 Hidden Layers
    - 4 Nodes



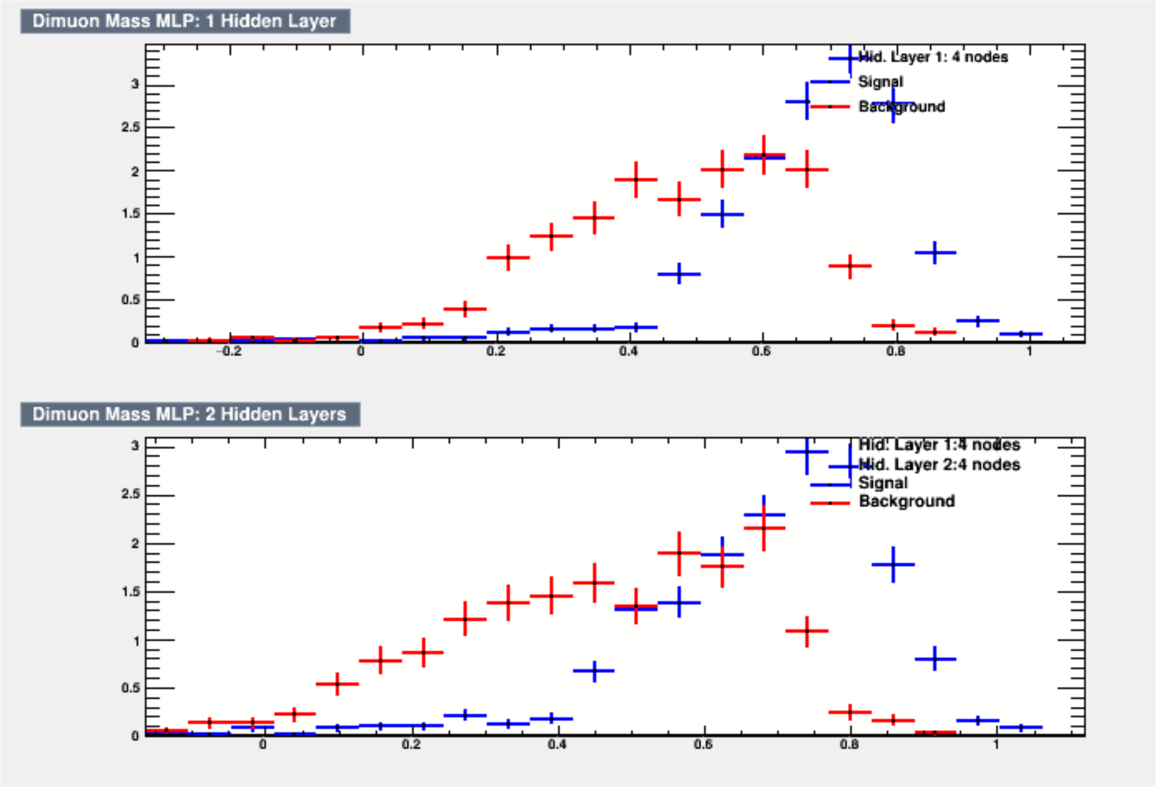
Classification (Case 2)

- Signal and Background
  - Signal input: Kinematic Variables in Dimuon Mass Peak
    - 80-100 GeV (1,758 events)
  - Background input: Surrounding Drell-Yan Mass Peak
    - 100-120 GeV (1,275 events)



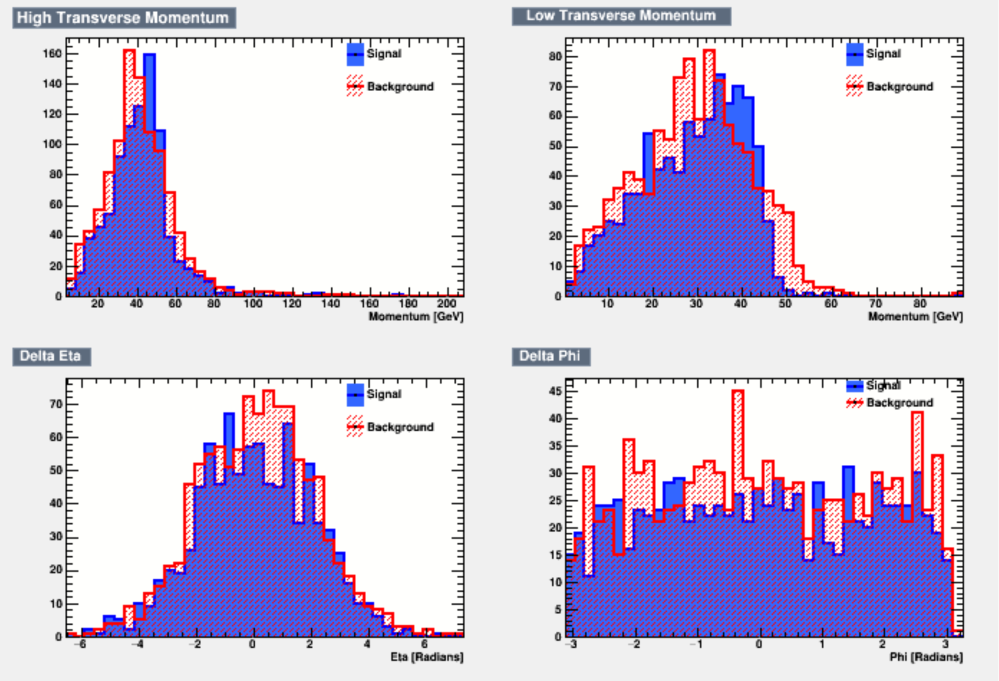
Classification (Case 2)

- MLP Attributes
  - Sigmoid activation function
  - 5,000 training cycles
  - 2 MLPs
    - 1 and 2 Hidden Layers
    - 4 Nodes



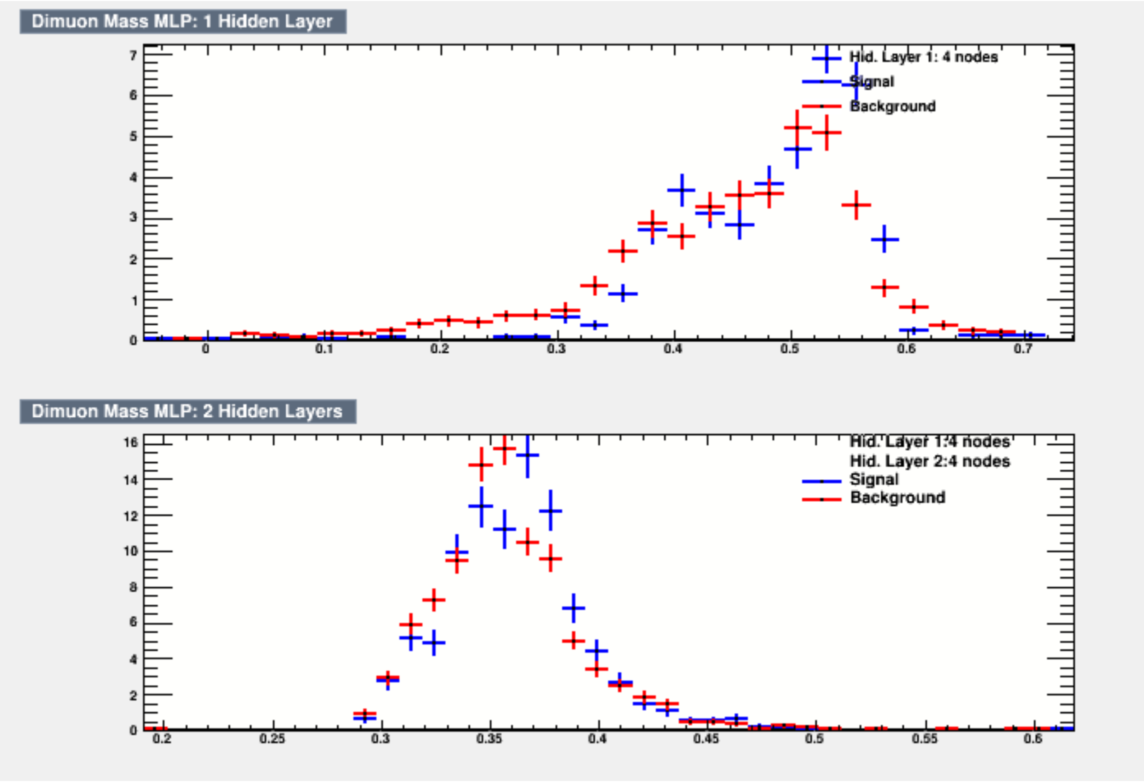
Classification (Case 3)

- Signal and Background
  - Signal input: Dimuon Mass Peak
    - 80-100 GeV (1,758 events)
  - Background input: Surrounding Drell-Yan Mass Peak
    - 60-80 GeV (1,867 events) and 100-120 GeV (1,275 events)



Classification (Case 3)

- MLP Attributes
  - Sigmoid activation function
  - 5,000 training cycles
  - 2 MLPs
    - 1 and 2 Hidden Layers
    - 4 Nodes

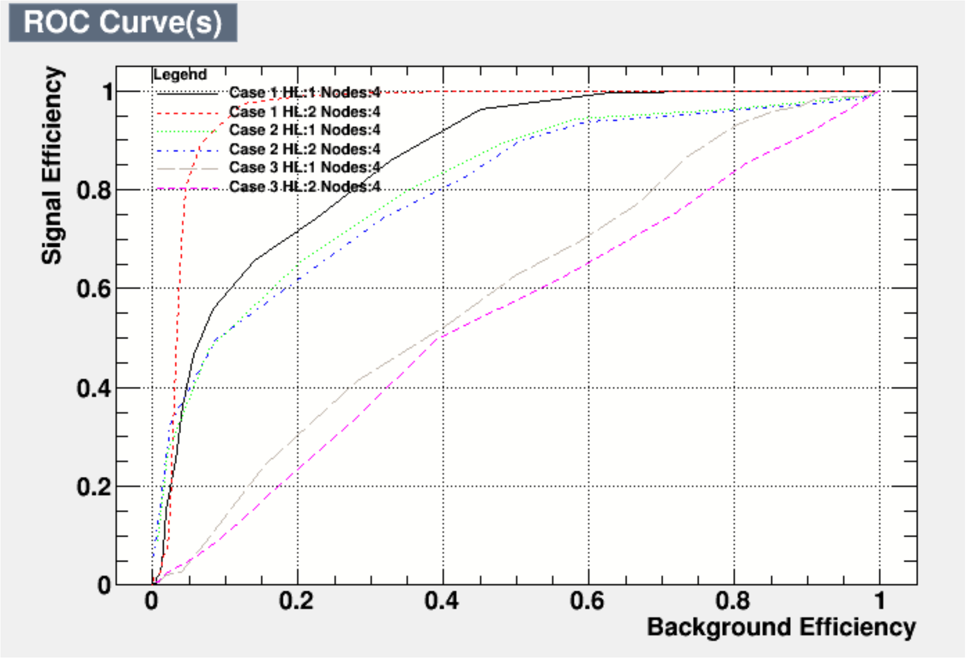


MLP Analysis of Dimuon Mass

Dimuon Mass MLP Attributes Table (Case 1)			
Hidden Layers:	Nodes in Layer One:	Nodes in Layer Two:	AUC
One	Four	N/A	0.8607
Two	Four	Four	0.9577

Dimuon Mass MLP Attributes Table (Case 2)			
Hidden Layers:	Nodes in Layer One:	Nodes in Layer Two:	AUC
One	Four	N/A	0.809
Two	Four	Four	0.797

Dimuon Mass MLP Attributes Table (Case 3)			
Hidden Layers:	Nodes in Layer One:	Nodes in Layer Two:	AUC
One	Four	N/A	0.595
Two	Four	Four	0.543



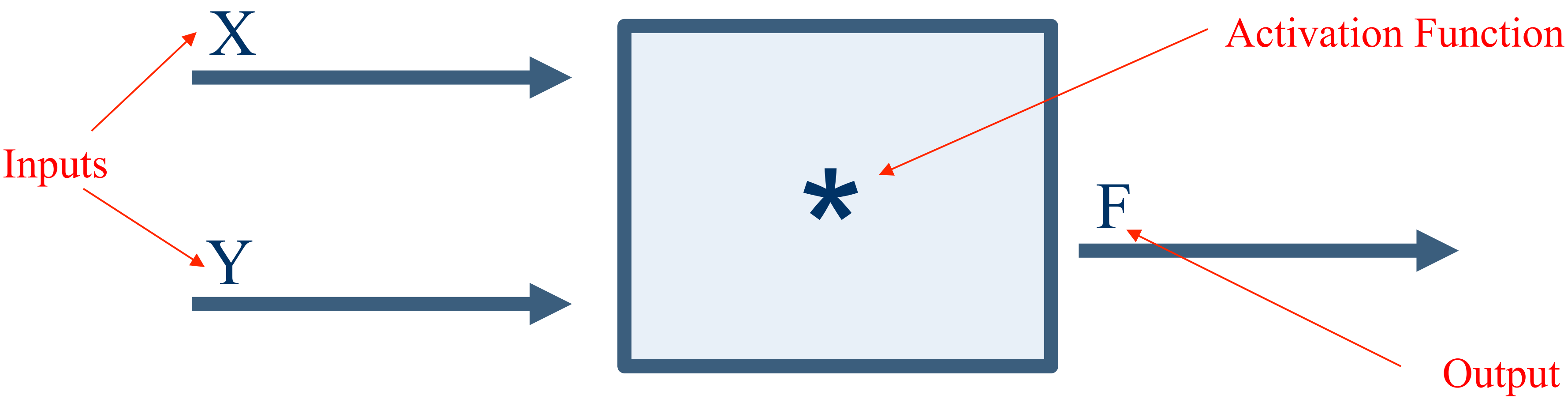
Summary

- Neural Networks can be used to separate signal and background events in collisions at CMS
- We used TMVA to apply neural networks in dimuon final state events
  - Applied to $Z \to \mu\mu$ and Drell-Yan
- Deeper MLPs do not increase separation much in these examples

Acknowledgements

- Advisors: Dr. Pushpa Bhat and Dr. Leonard Spiegel
- Co-workers: Graham Stoddard (Northern Illinois University) and Grace Cummings (Virginia Commonwealth University)
- Mentor: Dr. Elliot McCrory
- Staff: Sandra Charles and the SIST Committee

Simple Example



- A neuron can be thought of as a real-valued circuit
- Goal: slightly tweak input to increase the output
- Find $\partial F/\partial X$ and $\partial F/\partial y$
- Change each by a small step size
  - X' = X + step * $\partial F/\partial x$
  - Y' = Y + step * $\partial F/\partial Y$

Advances in Neural Networks; Deep Learning

- Deep Neural Networks
  - Very difficult to train in the past due to many layers
  - Recent advances in many research labs have made it easier to train
  - Relatively new Python modules such as Theano and PyLearn2 provide necessary framework for new DN studies
- Deep Belief Networks
  - Greedy layer-wise unsupervised learning
- Self-Organizing Maps
- Check out LISA Lab at University of Montreal for interesting work regard Deep Learning